Deploying a Next.js App with API on IIS (Windows Server)

1. Requirements

- Node.js installed on the Windows Server
- IIS with ARR (Application Request Routing) and URL Rewrite modules installed
- Next.js app built using the App Router and 'src' folder structure with API routes
- Administrative access to run PowerShell commands and configure IIS

2. Build Your Next.js App

Run these commands in your project root:

```
npm install
npm run build
```

This creates a `.next` folder with production-ready files.

3. Run Your Next.js App

Run the app on a specific port (default is 3000):

```
npm start
```

For persistent running, use PM2:

```
npm install -g pm2
pm2 start npm --name "nextjs-app" -- start
pm2 save
pm2 startup windows
```

4. Configure IIS as Reverse Proxy

1. Enable Application Request Routing (ARR) in IIS:
   - Open IIS Manager > Server Node > Application Request Routing > Enable Proxy

2. Create a new site in IIS:
   - Site Name: NextJsApp
   - Physical Path: C:\inetpub\wwwroot\NextJsDummy
   - Port: 80 (or custom)

3. Enable URL Rewrite and add a reverse proxy rule to redirect traffic to localhost:3000.

5. PowerShell Script to Automate Setup

Use the following PowerShell script (run as Administrator):

```
# Create IIS Site and Reverse Proxy Configuration

$siteName = "NextJsApp"
$sitePort = 80
$nodePort = 3000
$sitePath = "C:\inetpub\wwwroot\NextJsDummy"

if (-Not (Test-Path $sitePath)) { New-Item -ItemType Directory -Force -Path $sitePath }
Import-Module WebAdministration
New-Website -Name $siteName -Port $sitePort -PhysicalPath $sitePath -Force

Set-WebConfigurationProperty -filter "system.webServer/proxy" -name "enabled" -value "true" -pspath 'MACHINE/WEBROOT/APPHOST'

$webConfigContent = @"
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
```

```
        <rule name="ReverseProxyInboundRule1" stopProcessing="true">
          <match url="(.*)" />
          <action type="Rewrite" url="http://localhost:$nodePort/{R:1}" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
"@

$webConfigContent | Out-File -Encoding utf8 -FilePath "$sitePath\web.config"
```

6. Production Environment Variables

Create a `.env.production` file with necessary environment variables like:

```
PORT=3000
NEXT_PUBLIC_API_URL=https://your-api-url.com
NODE_ENV=production
```

7. Access Your App

Now you can access your app through IIS:

http://localhost:80

Or through your domain if DNS is configured.